

Sigle : INF1573 Gr. 01

Titre : Programmation II

Session : Hiver 2021 Horaire et local

Professeur : Benyahia, Ilham

1. Description du cours paraissant à l'annuaire :

Objectifs

Approfondir les concepts de la programmation orientée-objet. Sensibiliser au développement de programmes de qualité.

Contenu

Types abstraits, polymorphisme, généricité, événements. Utilisation de bibliothèques (JDK, STL ou similaire). Récursivité. Vérification, tests et documentation de programmes. Présentation des langages C et C++, discussion de certains concepts dans le contexte du langage Java : passage de paramètres, pointeurs, structures, "templates", etc. Notions d'analyse numérique: précision. Ce cours comporte des séances obligatoires de travaux dirigés (TD) de deux heures par semaine.

Descriptif – Annuaire

2. Objectifs spécifiques du cours :

1. Maîtriser les concepts de la programmation objet et leurs applications
2. Être capable d'analyser les conséquences d'utilisation des concepts objet
3. Acquérir des connaissances sur les composantes JDK permettant de faire la programmation par événements
4. Acquérir des expériences pour développer des interfaces graphiques basées sur la programmation par événements
5. Acquérir des connaissances sur les critères de qualité de la programmation par objet
6. Expérimenter les méthodes d'analyse de qualité de la programmation incluant l'application des concepts objets et leurs impacts
7. Distinguer la différence de la résolution de problèmes par les méthodes itératives et les méthodes récursives
8. Maîtriser l'application des concepts objets sur deux langages différents : JAVA et C++

Ce cours couvre 3 des 12 qualités requises des diplômé(e)s telles que définies dans les normes d'agrément des programmes de génie au Canada (<http://www.engineerscanada.ca/fr/ressources-en-matiere-dagrément>) :

a. Qualité 1 : Connaissance en génie

b. Qualité 2 : Analyse de problèmes

c. Qualité 5 : Utilisation d'outils d'ingénierie

Les qualités 2 et 5 sont mesurées dans ce cours pour fins de rétroaction.

Objectifs spécifiques	Qualité	Indicateurs	Introduit	Développé	Utilisé
1. Maîtriser les concepts de la programmation objet et leurs applications.	2	2. Formuler un processus de résolution de problème, comprenant des approximations et des hypothèses.		x	
2. Être capable d'analyser les conséquences d'utilisation des concepts objets.	2	4. Évaluer les résultats obtenus et formuler des conclusions.		x	

1. Acquérir des expériences pour développer des interfaces graphiques basées sur la programmation par événements.	5	2. Utiliser les outils, techniques de mesure, modèles ou simulations appropriés.		x	
---	---	--	--	---	--

3. Stratégies pédagogiques :

Les formules pédagogiques suivantes seront utilisées :

L'enseignement est dispensé sous forme magistrale (3 h/semaine) en non-présentiel en mode synchrone via la plateforme Zoom. La stratégie pédagogique est basée sur une formation active nécessitant des interactions de manière individuelle avec la professeure ou par des activités en groupe. Des exercices seront donnés durant ces séances pour illustrer les notions nouvellement introduites.

Les étudiant(e)s doivent participer en classe pour mieux saisir les notions qui seront présentées pendant le cours. En plus des travaux pratiques évalués, d'autres travaux seront demandés pour renforcer l'atteinte des objectifs du cours.

Des séances de travaux dirigés (2 h/séance) en non-présentiel viendront appuyer les séances de cours. Ces séances seront supervisées à distance par un assistant d'enseignement. La présence des étudiant(e)s est obligatoire.

L'ensemble des évaluations auront lieu en non-présentiel (remise de travaux selon les délais qui seront spécifiés dans les énoncés avec une présentation orale via Zoom pour le projet de session).

Remarques :

- Les modalités de cours et d'évaluation sont sujettes à modification selon l'évolution de la situation sanitaire.
- Les étudiant(e)s qui s'inscrivent à ce cours doivent s'assurer qu'ils disposent d'un ordinateur avec webcam et microphone, et d'une connexion Internet. Ils doivent pouvoir installer certains logiciels gratuits. Les détails sur ces installations seront donnés sur le site Moodle du cours.

[Guide d'utilisation de Zoom à l'intention des étudiants](#)

Site pour soutien de réussite en mode non-présentiel : uqo.ca/etudier-non-presentiel.

4. Heures de disponibilité ou modalités pour rendez-vous :

Consultations via la plateforme Zoom : sur rendez-vous. ilham.benyahia@uqo.ca

5. Plan détaillé du cours sur 15 semaines :

Semaine	Thèmes	Dates
1	Introduction Plan du cours Présentation des concepts objets : <ol style="list-style-type: none"> 1. Concepts de classes et objets 2. Concepts d'héritage et polymorphisme 3. Concepts d'encapsulation Notions de généricité Notions sur les types abstraits	13 janv. 2021

2	<p>Programmation orientée objet : Différentes formes d'héritage et notions sur la réutilisation</p> <ul style="list-style-type: none"> • Introduction • Relation d'héritage • Étude de cas <p>Travail dirigé #1 : Le jeudi 21 janvier ou le mardi 26 janvier 2021 : Les concepts objets</p>	20 janv. 2021
3	<p>Programmation orientée objet (suite) : Polymorphisme</p> <ul style="list-style-type: none"> • Introduction • Classes et méthodes abstraites • Études de cas • Test formatif 1 <p>Travail dirigé #2 : Le jeudi 28 janvier ou le mardi 02 février 2021 : Les concepts objet (Suite – TD 1)</p>	27 janv. 2021
4	<p>Programmation par événements</p> <ul style="list-style-type: none"> • Les composantes graphiques • Les Applets • Les swings • Études de cas <p>Travail dirigé #3 : Le jeudi 04 février ou le mardi 09 février 2021 : Applications : relations entre les classes et héritage</p>	03 févr. 2021
5	<p>Principes de qualité de la programmation</p> <ul style="list-style-type: none"> • Introduction • Tests de vérification • Énoncé du devoir 1 <p>Travail dirigé #4 : Le jeudi 11 février ou le mardi 16 février 2021 : Utilisation du polymorphisme</p>	10 févr. 2021
6	<p>Qualité de la programmation (suite)</p> <ul style="list-style-type: none"> • Étude de cas • Révision pour l'examen intra • Projet de session <p>Travail dirigé #5 : Le jeudi 18 février ou le mardi 23 février 2021 : La programmation par événements</p>	17 févr. 2021

7	Qualité de la programmation (suite) <ul style="list-style-type: none"> • Documentation des programmes Travail dirigé #6 : Le jeudi 25 février ou le mardi 09 mars 2021 : Analyse de la qualité de la programmation	24 févr. 2021
8	Semaine d'études	03 mars 2021
9	Le traitement des exceptions <ul style="list-style-type: none"> • Bases de traitements des exceptions • Exemples de classes d'exceptions • Déclarations de nouveaux types d'exception Travail dirigé #7 : Le jeudi 11 mars ou le mardi 16 mars 2021 : Exceptions en java	10 mars 2021
10	Le traitement des exceptions (suite) <ul style="list-style-type: none"> • Test formatif 2 Travail dirigé #8 : Le jeudi 18 mars ou le mardi 23 mars 2021 : Applications des exceptions	17 mars 2021
11	Récurtivité <ul style="list-style-type: none"> • Concepts de récursivité • Exemples utilisant la récursivité • Récursivité vs itération Travail dirigé #9 : Le jeudi 25 mars ou le mardi 30 mars 2021 : Résolution de problèmes en java par la récursivité	24 mars 2021
12	Étude de cas <ul style="list-style-type: none"> • Résolution de problèmes d'analyse numérique • Énoncé du devoir 2 Travail dirigé #10 : Le jeudi 01 avril ou le mardi 06 avril 2021 : Résolution de problèmes en java	31 mars 2021
13	Java vs C++ : héritage multiple, amies, fonctions virtuelles, surcharge des opérateurs, gabarits, pointeurs	07 avril 2021
14	Présentations des résultats du projet de session	14 avril 2021
15	Examen final en non-présentiel	21 avril 2021

6. Évaluation du cours :

Outils d'évaluation	Pondération	Indicateurs mesurés
Projet de session	35 %	5.2, 2.2 et 2.4
Devoir 1 et Devoir 2	30 %	2.2 et 2.4
Examen final	30 %	2.2 et 2.4
Travaux dirigés	5 %	

Par **indicateur mesuré**, on entend qu'à la fin du cours, un niveau de performance (0, 1, 2, 3) est donné pour chaque indicateur et pour chaque étudiant(e) selon la grille ci-dessous :

Indicateurs	Niveau 0	Niveau 1	Niveau 2	Niveau 3
2.2 – Formuler un processus de résolution de problème, comprenant des approximations et des hypothèses.	Formulation du processus de résolution inacceptable et traitement inadéquat des approximations et des hypothèses	Formulation du processus de résolution acceptable, mais traitement partiel des approximations et des hypothèses	Formulation du processus de résolution et traitement des approximations et des hypothèses acceptables	Formulation du processus de résolution et traitement des approximations et des hypothèses remarquables
2.4 – Évaluer les résultats obtenus et formuler des conclusions.	Évaluation et/ou conclusions inexistantes	Évaluation et conclusions partielles	Évaluation et conclusions acceptables	Évaluation et conclusions remarquables
5.2 – Utiliser les outils, techniques de mesure, modèles ou simulations appropriés.	Utilisation inadéquate ou inexistante	Utilisation partielle	Utilisation adéquate	Utilisation remarquable

7. Politiques départementales et institutionnelles :

- Politique du département d'informatique et d'ingénierie relative à la tenue des examens
- Note sur le plagiat et sur la fraude
- Politique relative à la qualité de l'expression française écrite chez les étudiants et les étudiantes de premier cycle à l'UQO
- Absence aux examens : cadre de gestion, demande de reprise d'examen (formulaire)

À l'UQO, **les violences à caractère sexuel, c'est tolérance zéro!**

La communauté universitaire s'engage à lutter contre les inconduites, le harcèlement et les violences à caractère sexuel : parce que **le respect, c'est l'affaire de tout le monde!**

N'oubliez pas de faire la formation obligatoire :

uqo.ca/bimi/formation-obligatoire

Pour de plus amples renseignements :

bimi@uqo.ca



8. Principales références :

Les notes de cours représentent la principale référence. Les étudiant(e)s doivent aussi consulter régulièrement des documents parmi ceux qui seront suggérés et particulièrement ceux qui seront mis en réserve à la bibliothèque.

Non obligatoires mais fortement recommandés :

- Object-Oriented Problem Solving. Java, Java, Java. Ralph Morelli, Prentice-Hall. 2003, ISBN 0-13-033370-0.
- Java How to program, 6th Edition. H. M. Deitel et P.J. Deitel. Prentice Hall, 2005. ISBN 0-13-148398-6.

Autres références – à consulter :

- On To Java. P. H. Winston et S. Narasimhan, 3rd Edition (version en ligne).
- Object-Oriented Programming with Java – An Introduction, D. J. Barnes, Prentice-Hall, 2000, ISBN 0-13-086900-7.
- Understanding Object-Oriented Programming with Java. T. Budd, Addison-Wesley, 2000, ISBN 0-201-61273-9.
- An Introduction to Object-Oriented Programming third edition, T. Budd, Addison-Wesley. 2001, ISBN: 0-201-76031-2.
- The Java Class Libraries Second Edition, Volume 1. P. Chan, R. Lee and D. Kramer. Addison-Wesley, 1998, ISBN:0 201-31002-3.
- Object-Oriented Programming with C++ and Smalltalk, Caleb Drake, Prentice-Hall. 1998, ISBN: 0-13-103797-8.

9. Page Web du cours :

<https://moodle.uqo.ca>